

Junger, Christina; Heß, A.; Rosenberger, Maik; Notni, Gunther:

FPGA-accelerated phase rectification for a stereo-based phase measuring profilometry system

Original published in: Journal of physics. Conference Series / Institute of Physics - Bristol : IOP Publ.. - 1065 (2018), art. 32017, 4 pp.
Original published: 2018-08-01
ISSN: 1742-6596
DOI: [10.1088/1742-6596/1065/3/032017](https://doi.org/10.1088/1742-6596/1065/3/032017)
[Visited: 2020-06-09]



This work is licensed under a [Creative Commons Attribution 3.0 Unported](https://creativecommons.org/licenses/by/3.0/) license. To view a copy of this license, visit <https://creativecommons.org/licenses/by/3.0/>

FPGA-accelerated Phase Rectification for a Stereo-based Phase Measuring Profilometry System

C Junger¹, A Heß, M Rosenberger and G Notni

Research Assistant, Technische Universität Ilmenau,

Group for Quality Assurance and Industrial Image Processing, 98693 Ilmenau, DE

Email: christina.junger@tu-ilmenau.de

Abstract. This paper proposes a FPGA-accelerated lens undistortion and rectification algorithm for a stereo-based phase measuring profilometry system. After a brief system overview we go into detail on the proposed hardware architecture for the stereo rectification block. The hardware architecture is based on a Xilinx Zynq-7020 SoC and uses compressed rectification maps to reduce the memory load. As a result, it is able to rectify a stereo setup with 435 mm baseline and converged optical axis of 32° between the cameras. The cameras are configured with 2 Mpix @ 60 fps and rectification can be applied “on the fly” during image grabbing.

1. Introduction

Phase Measuring Profilometry (PMP) is a widely used technique in structured light 3D shape measurement [1]. There are several application fields for structured light scanners, such as quality control or medical science. Image processing algorithms have to be accelerated due to the need of high speed inline measurement systems. In embedded environments FPGAs should be a suitable solution for this task. [2]

2. Theoretical Background and relevant Research

2.1. Phase measuring profilometry

PMP uses multiple phase-shifted sinusoidal fringe patterns which are projected onto the measuring surface. The patterns have the same shape and are shifted by a constant angle. The phase information $\phi(x, y)$ can be determined from the intensities of the projected fringe patterns using equation 1. m is the number of phase shifts within a 2π period and i the shifted pattern index [3].

$$\phi(x, y) = \tan^{-1} \frac{-\sum_{i=1}^m I_i(x, y) \sin\left[\frac{2\pi}{m}(i-1)\right]}{\sum_{i=1}^m I_i(x, y) \cos\left[\frac{2\pi}{m}(i-1)\right]} \quad (1)$$

In common practice four patterns shifted by $\pi/2$ are used to determine the relative phase. Twelve patterns should be used to achieve highest accuracy [4]. Equation 1 can be solved using the four quadrant arctangent function atan2 that determines values in the interval of $[-\pi, \pi)$, that are only relative phase values within a sinusoidal period. The so called wrapped phase image has to be unwrapped to form a continuous phase map and solve the problem of global ambiguities that are problematic for phase comparison in the stereo image. A common method for phase unwrapping is to project a sequence of

¹ To whom any correspondence should be addressed.



grey codes subsequently to the sinusoidal patterns [1]. The grey code is binary coded in such a way, that every period in sinusoidal patterns gets its own unique grey code index. This index is used to add multiples of 2π to the relative phase image. The combination of sinusoidal patterns and grey codes leads to a continuous phase image that is suitable for phase comparison and disparity calculation. However, before the phase matching starts, the phase images of both cameras should be rectified making them line correspondent.

2.2. Lens undistortion and phase image rectification

On FPGA-based SoCs it is common practice to realize the lens undistortion and image rectification with an inverse mapping algorithm [5],[6],[7]. In a stereo setup an undistortion and rectification transformation map has to be generated for each camera. This is commonly done by OpenCV functions [8]. By applying the undistort and rectify maps to the camera images, lens undistortion and stereo rectification is performed in one step. The map generation has to be done only once as long as the stereo geometry and camera optics remain unchanged. The calibration algorithm assigns a subpixel accurate pixel position within the distorted image for each pixel position in the rectified image. Then a bilinear interpolation is performed for calculating the new grey value. [2],[5] After the image rectification the search for corresponding points between the two cameras is defined as a search task along corresponding image lines. Two undistortion and rectification transformation maps must be stored per camera. One of these maps includes x-offset values and the other y-offset values. Storing these values as float or double will accordingly cause a higher memory load as well as a higher utilization of bandwidth.

3. Proposed system and hardware architecture

3.1. System overview

Our proposed system is based on a Xilinx Zynq-7020 SoC, Basler dart cameras with BCON interface and a LC4500 industrial DLP projector from *Texas Instruments*. The two cameras are equipped with e2V EV76C570 CMOS sensors, which provide framerate up to 60 fps @ 2 Mpix resolution. Figure 1 shows an overview of the system comprised of a hardware part with programmable logic cells (Zynq PL) and a software part with an ARM Cortex A9 MP Core running *Linux* (Zynq PS) with a shared

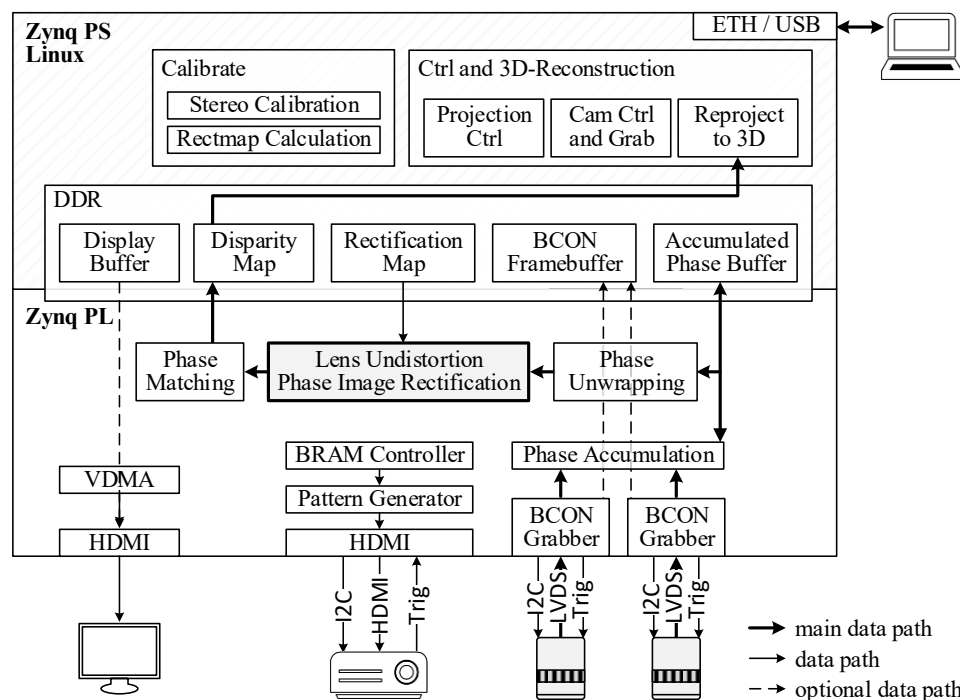


Figure 1. System overview based on Zynq-7020 SoC, comprising a programmable logic part (Zynq PL) and a processing system running Linux (Zynq PS)

memory in between. Camera calibration and 3D reconstruction are implemented by standard OpenCV functions [8]. Phase accumulation, phase unwrapping, lens undistortion, phase image rectification and phase matching are implemented in the PL. These are function blocks that are convenient for hardware acceleration. In a typical measuring procedure, the phase images are accumulated by PL in the external memory during image grabbing. Equation 1 shows that nominator and denominator can be accumulated separately from each other independent of the number of fringe images. This means that the accumulated image is read out synchronously to the image grabber and the new accumulation result is written back to memory. In this manner the unwrapping of the relative phase can start with the last fringe pattern image that is grabbed by the cameras. After capturing the grey codes they will be used to remove the discontinuities in the relative phase image by adding multiples of 2π (subsection 2.1). After unwrapping the phase image for each camera, they need to be rectified according to the stereo geometry so that phase matching is defined as a search task along corresponding camera lines. The phase image module then generates a disparity map by finding corresponding phase values between the two unwrapped phase images. Phase matching, as well as the other functions, can be fully pipelined by parallelization.

3.2. Lens undistortion and phase image rectification

As already mentioned in subsection 2.2, undistortion and rectification transformation maps (UndistRectMaps) are calculated offline. Each camera needs two maps which contain the absolute horizontal and vertical pixel positions. Due to the need of storing the maps in external memory and loading them simultaneously to the phase images, a compression is needed so that the memory bandwidth is not so heavily utilized. Thus, redundant information must be removed. At first, the compression algorithm subtracts the pixel index so the maps contain only the relative pixel offsets. In the next step the offset values are quantized to seven binary decimal places. To compress the maps even further the offset differences are calculated by the first derivation of a row. In the compressed result the absolute pixel offset values (X and Y) are stored in the first column, the remaining columns contain only the offset differences. For optimizing the memory access, the horizontal and vertical maps for each camera need to be merged. Thus, horizontal and vertical offset-diffs (x and y) are column-wise alternating in one map. The differences are stored with a sign bit in addition to the seven binary decimal places. This results in a final UndistRectMap size of $2 \cdot N \times M$ bytes (Figure 2).

Figure 2 shows the hardware architecture of the lens undistortion and phase image rectification unit (LDRU) for one camera. The bold arrows show the main path of the phase image. Simple dual port RAMs are used for buffering even and odd rows of the phase image separately. The rectification maps are loaded by DMA from external memory. It is obvious that they have to be synchronized to the buffered phase image data stream. Accumulators are used to reconstruct the current horizontal and vertical offsets from the offset-diffs. The integer and decimal parts are treated separately, since the

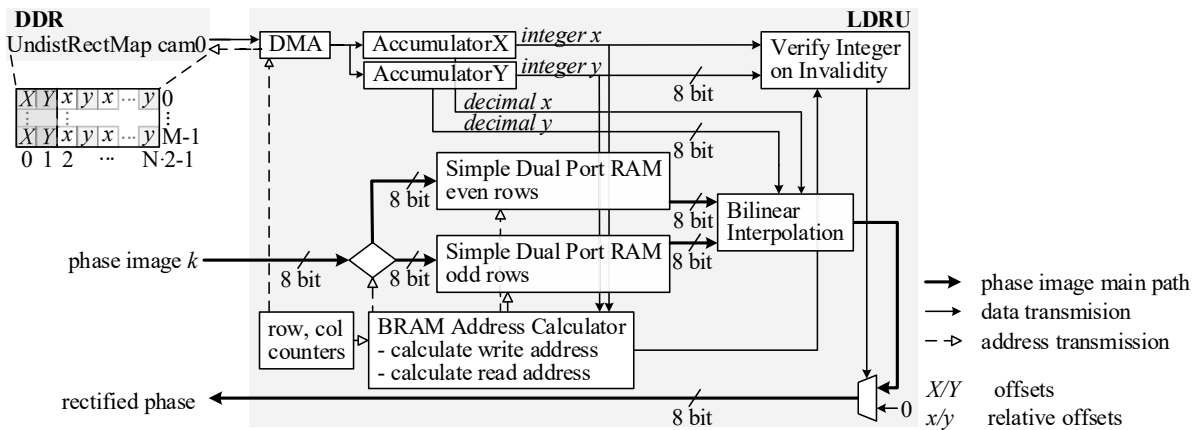


Figure 2. Overview of the lens undistortion and phase image rectification unit (LDRU) for one camera

BRAM address calculator uses the two integer parts to determine the corresponding position of the four grey values in the two BRAMs. With these grey values and the two decimal parts, a bilinear interpolation is performed. It is necessary to test each integer on invalidity. Valid integers have to be within the range of image size ($M \times N$) and invalid integers will generate a zero in the rectified phase image.

4. Conclusions and future work

The proposed measurement system is configured with a base line of 435 mm and converged optical axis of 32° between the cameras. The generated undistortion and rectification transformation maps (UndistRectMaps) have maximum offsets of ± 48 pixel. The proposed LDRU with compressed UndistRectMaps requires only a quarter of the memory load that would be utilized by OpenCV generated UndistRectMaps (Table 1). With a camera resolution of 1600×1200 pixel 21 BRAM blocks are needed for each the even and the odd rows buffer, in Zynq-7020 SoC. The LDRU utilizes 30 % of the available BRAM resources [10]. In comparison to the use of uncompressed UndistRectMaps the rectified image shows a deviation of one grey value at the most (along strong gradients) using a binary precision of seven decimal places for the offset values. This was confirmed experimentally.

It is necessary to further compress the UndistRectMaps in order to further reduce the bandwidth. One conceivable aid would be the subsampling of the UndistRectMaps. However, it should be noted that this can lead to further inaccuracies.

Table 1. Difference in memory utilization between OpenCV and compressed UndistRectMaps

UndistRectMap for one camera	Map type	Data type	Size (byte)	Memory load (MB/s) @ 2 Mpix, 60 fps	Bandwidth utilization ^a (%)
OpenCV	x	float	$(M \times N) \cdot 4$	960	22,5
	y	float	$(M \times N) \cdot 4$		
Compressed	merged	unsigned short	$(M \times N) \cdot 2$	240	5,6

^a Zynq-7000 32-bit DDR3 memory controller: maximal theoretical bandwidth 4267 MB/s [9]

Acknowledgments

The research presented in this paper is funded by the Free State of Thuringia, the European Social Fund (ESF) of the European Union, the Thüringer Aufbaubank (TAB) (2016 FGR 0044) and the German Federal Ministry of Education and Research (BMBF, FKZ: 03ZZ0442E).



References

- [1] Geng J 2011 *Proc. SPIE* **7932** 79320B
- [2] Zhan G *et al* 2017 *J. Optics Express* **25** pp 10553-10564
- [3] Kreis T 1996 *Holographic interferometry: principles and methods (Akademie Verlag)* section 4.5 pp 123-138
- [4] Heist S *et al* 2015 *J. Applied Optics* **54** pp 10541-10551
- [5] Winkler S *et al* 2017 *59th Ilmenau Scientific Colloquium (Ilmenau, GER Sept 2017)* **59** 2.2.03
- [6] Staudinger E *et al* 2008 *Proc. FH Science Day 2008* pp 18-25
- [7] Mun J and Kim J 2015 Real-time fpga rectification implementation combined with stereo camera in *2015 IEEE International Symposium on Consumer Electronics (ISCE 2015)* **2015-08**
- [8] OpenCV 2011-2014 Camera calibration and 3d reconstruction **2.4.13.6**
- [9] Lucero J and Slous B 2014 Designing high-performance video systems with the Zynq-7000 all programmable SoC using ip integrator XAPP1205 **v1.0** pp 1-15
- [10] XILINX 2017 Zynq-7000 all programmable SoC overview DS190 **v1.11**